Hum 2017年07月12日

第二章：一个新的系统（LTF）

| 本文为付费栏目文章，您已订阅，可阅读全文 |

现在智能设备上的任务管理工具，从 iOS 内置的提醒事项到最复杂的 2Do、OmniFocus 以及我们这套教程的主角 Todoist。其实都在运用的同一套系统，这个系统的核心部分是**列表（List）**、**标签（Tag）**和**过滤（Filter）**，取其英文单词的首字母，可以简称为 LTF 系统。

但是因为大家对这个系统的思考不同，也许有的开发者都没有意识到有这么一个系统，所以这些工具对同样的功能实现的形式和程度也很不一样，名称差异也很大，取我眼中最有代表性的三个高级工具来对比：

后文中我会在相应位置详细说明这些功能，在这里只是想让大家意识到这些工具使用的都是这一个系统。一旦你意识到这一点，你在对比工具的时候就有了一个基础标准，有一个在比较时用于定位的东西。

项目

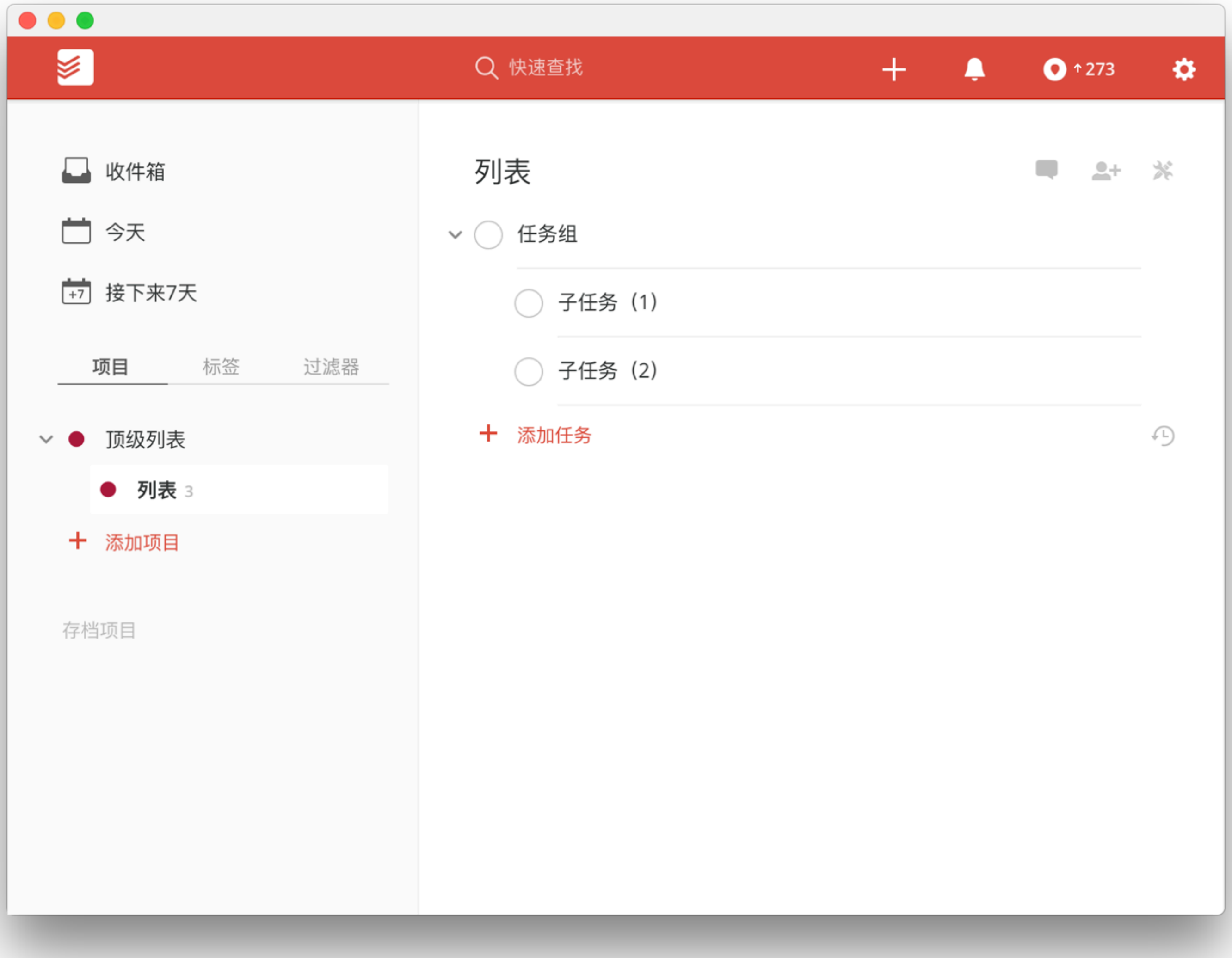
在任务管理工具里，所有的项目本质上都是列表，只是表现形式不同，以及因为表现形式不同而被赋予的内容不同。但原则上，只要是两个以上的东西以列表形式放在一起了，你就可以把它视为一个列表，比如说：

–1

–2

这就是一个列表。

在任务管理工具里，列表这个形式按层级可以分为三种：**顶级列表、列表和任务组**。

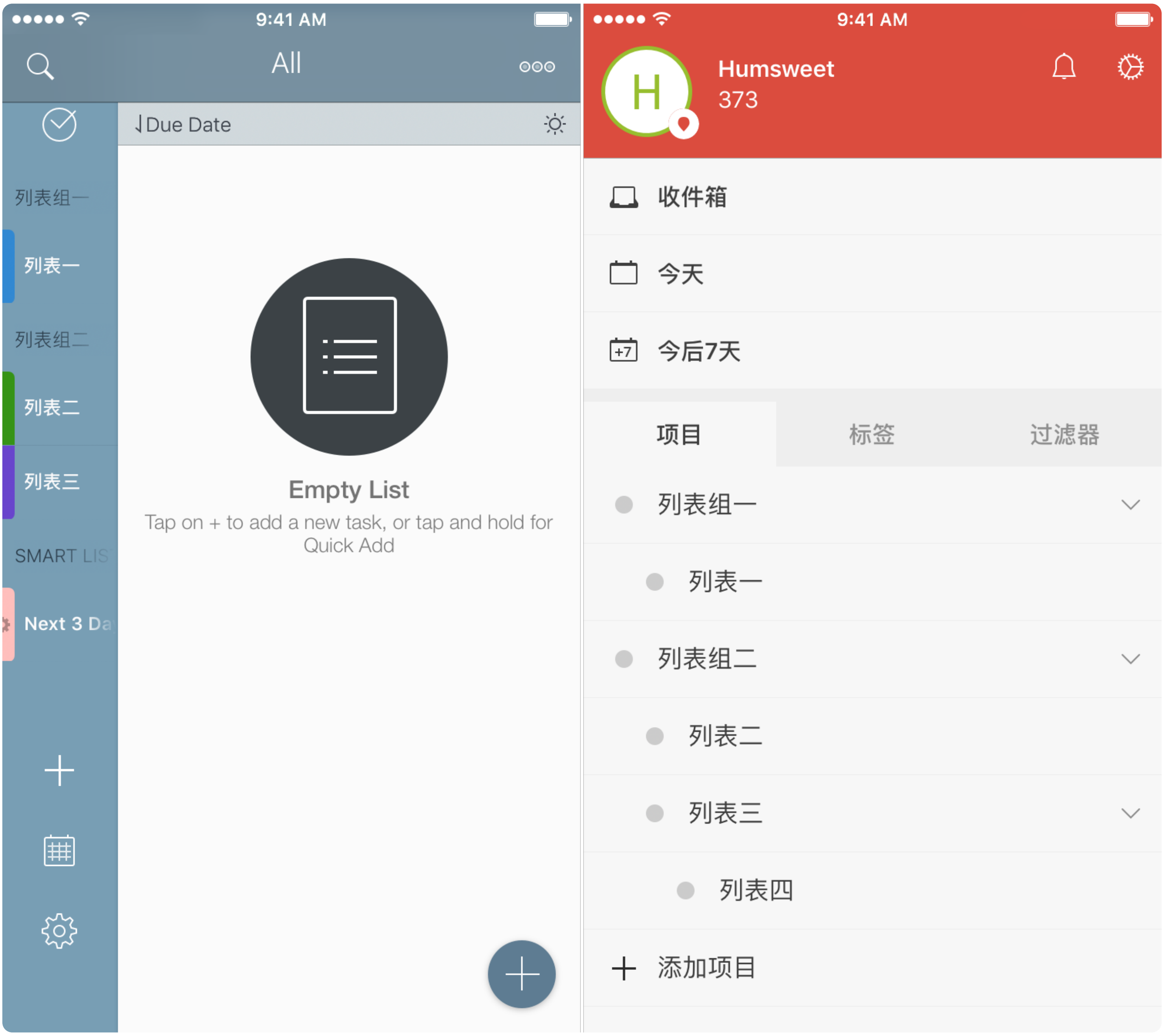

快速查找
273
收件箱
今天
接下来7天
项目
标签
过滤器
顶级列表
列表 3
添加项目
存档项目
列表
任务组
子任务（1）
子任务（2）
添加任务

顶级列表

不同于 OmniFocus 和 2Do，Todoist 的顶级列表本身就是列表，所以它既可以存放任务，也可以起到分隔列表的作用。

顶级列表在 OmniFocus 跟 2Do 里分别叫做**文件夹（Folder）**和**列表组（List Group）**，这两个顶级列表之下都只能放列表，而不能放任务。你可以理解为，它们是列表的容器而不是任务的容器，形式上起到分隔不同列表的作用。

但在 Todoist 里，顶级列表同列表本质上都是列表，只是层级不同，因此都可以放任务。换句话说，Todoist 里的顶级列表里既可以放列表，也可以放任务；既可以作为任务的容器，也可以起分隔列表的作用。


9:41 AM
All
↓Due Date
列表组一
列表一
列表组二
列表二
列表三
SMART LIS
Next 3 Da
Empty List
Tap on + to add a new task, or tap and hold for Quick Add
9:41 AM
H
Humsweet
373
收件箱
今天
今后7天
+7
项目
标签
过滤器
列表组一
列表一
列表组二
列表二
列表三
列表四
添加项目

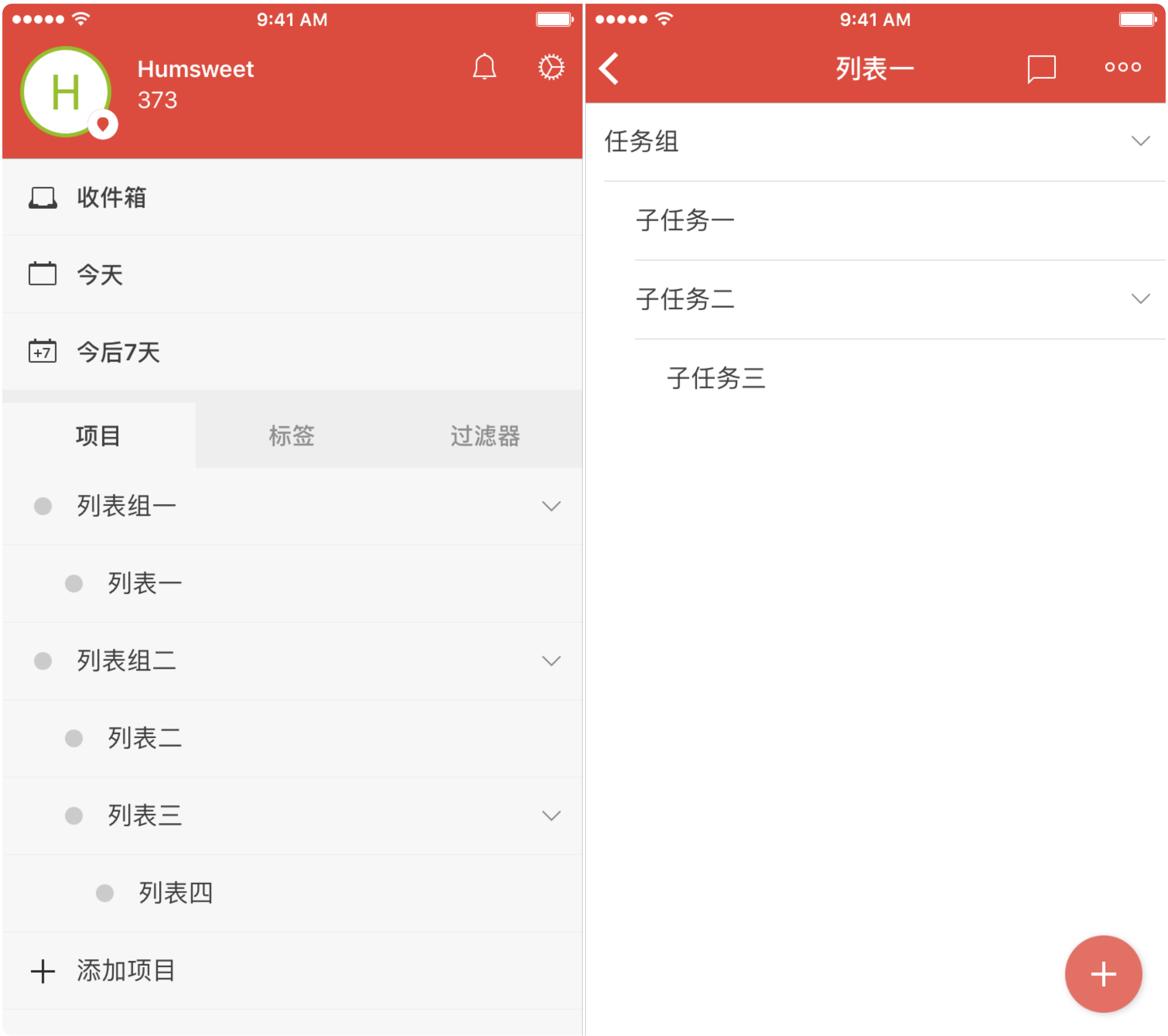
我个人是认为 Todoist 的形式更加灵活和现代。如果用纸笔来做任务管理的话，我们是会需要把任务写在纸上，每张纸形成一个任务清单，最后把多个清单放到文件夹里。而且我们一般不会直接在文件夹上写任务。但在电子系统里已经完全没必要这样做了。退一步讲，如果我们一定要在形式上抠得非常严谨的话，我们同样可以不在 Todoist 的顶级列表里添加任务。

总之，在 Todoist 中，顶级列表和列表仅是位置上的差异，没有功能上的差异，这就让我们在使用 Todoist 时更容易掌握列表的概念。

列表和任务组

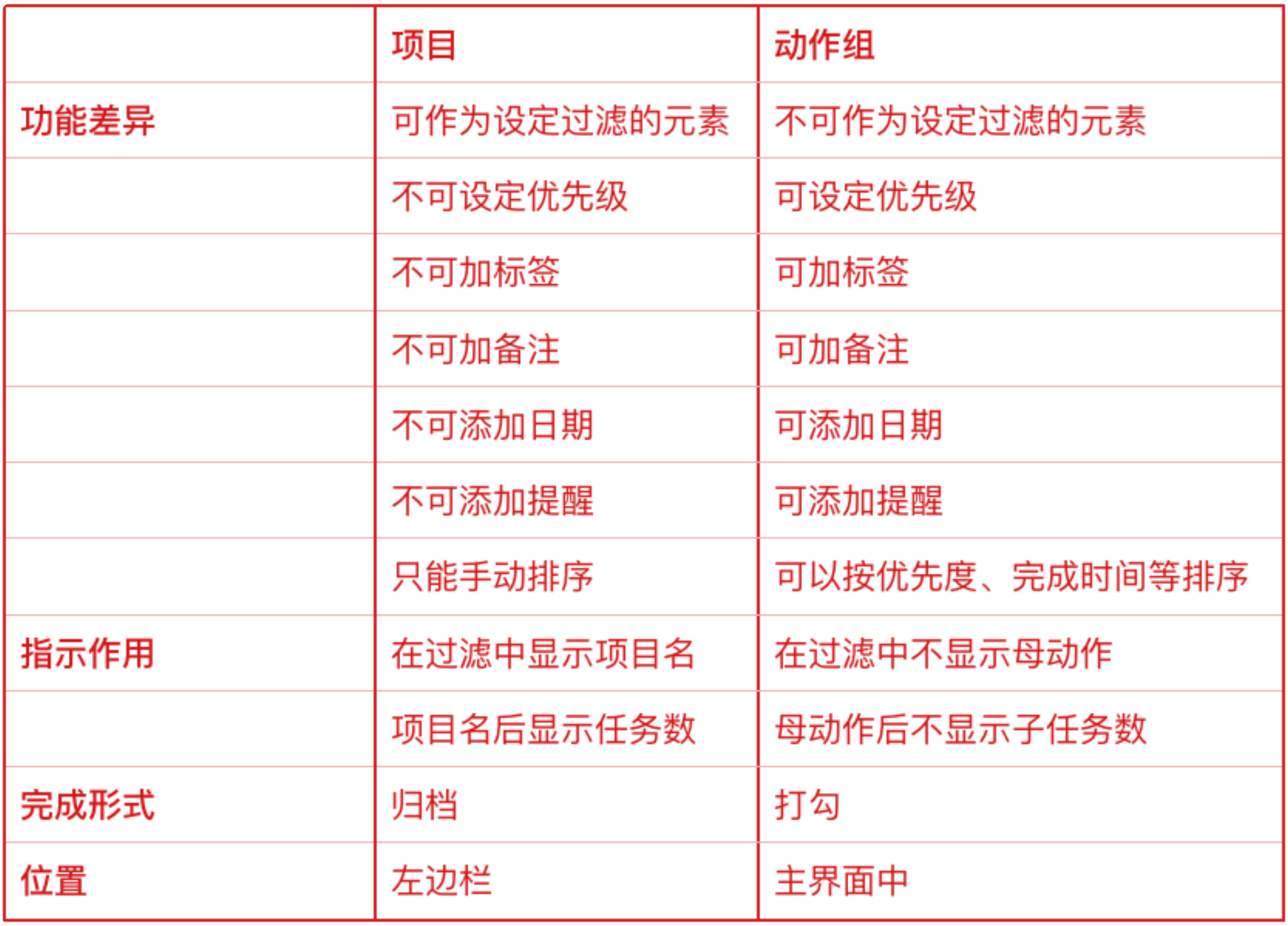
如果分不清该用列表还是列表组。要从实践的角度，从功能出发进行选择，不要太拘泥于形而上的定义。

不少人在使用这些高级工具的时候已经发现，我们不但可以在列表里添加任务，我们还可以在任务下添加子任务（Sub-tasks），来形成任务组。那我们应该如何去区分列表和任务组？以及什么时候该用列表，什么时候该用任务组？


9:41 AM
Humsweet
373
H
收件箱
今天
今后7天
+7
项目
标签
过滤器
列表组一
列表一
列表组二
列表二
列表三
列表四
添加项目
9:41 AM
列表一
任务组
子任务一
子任务二
子任务三
+

从定义上看：列表即项目，而任务只是项目的一部分，为了完成这个任务需要进一步做的事，则是这个任务的子任务。进一步地把定义抽象——可以理解为「列表包括任务，任务包括子任务。」而按照这个定义，列表和任务之间的界限就不是很清晰了。

所以我们在看任务管理工具的时候，要把实践意义放到定义之上。因为从定义上虽然不好判断，但作用到具体工具，列表和动作组的差异就很明显了。



	项目	动作组
功能差异	可作为设定过滤的元素	不可作为设定过滤的元素
	不可设定优先级	可设定优先级
	不可加标签	可加标签
	不可加备注	可加备注
	不可添加日期	可添加日期
	不可添加提醒	可添加提醒
	只能手动排序	可以按优先度、完成时间等排序
指示作用	在过滤中显示项目名	在过滤中不显示母动作
	项目名后显示任务数	母动作后不显示子任务数
完成形式	归档	打勾
位置	左边栏	主界面中

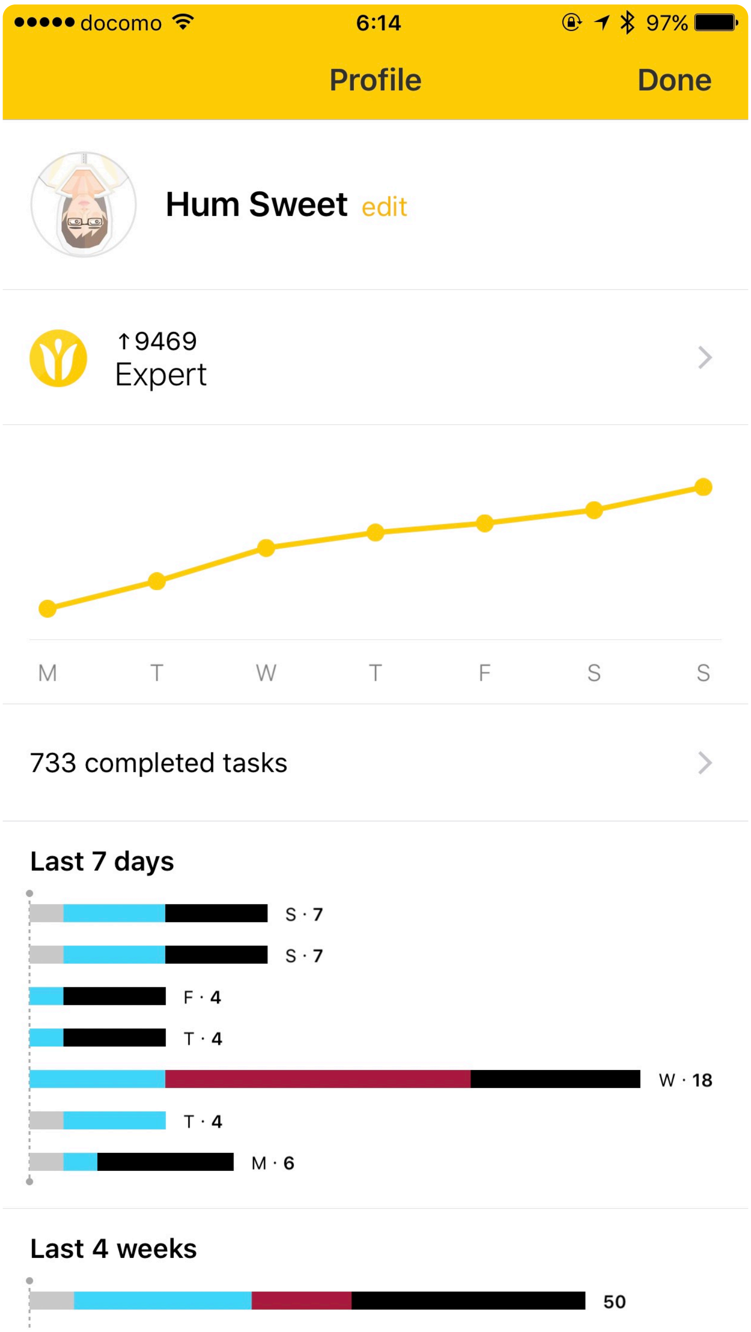
看完这个表格相信大家在 Todoist 中应该选择项目还是任务组就会有自己的主意了。在判断应该做项目还是做动作组的时候，只要参考自己最在乎的那一个要素就好。比如你非常需要一个任务在过滤中显示出它所属的那个项目，那你就没有别的选择，只能通过项目而不能通过任务组来实现它。

在任务管理中，**实践意义永远高于形而上的定义**。在看大家通过邮件反馈的问题时，我能看出有些读者对任务管理工具和理念都认真研究过，其结果是在形而上的定义与实际操作不一致之间纠结，反而看不见眼前的解决方案。而 Todoist 的好处就在这，它涉及的术语和理论框架非常少，用的时候可以尽管从实践入手，不必太在乎定义的问题。

项目的颜色选择

项目的颜色与在 Karma 界面下一览任务完成比例有关，所以建议每个大项保持一种颜色。

在 Todoist 里建立列表的时候你会发现它可以让你选择列表的颜色，喜欢用什么颜色当然还是看你个人喜好，但是这个颜色在这里除了美观之外还有一个实际作用：


●●●●● docomo
6:14
97%
Profile
Done
Hum Sweet
edit
↑9469
Expert
M
T
W
T
F
S
S
733 completed tasks
Last 7 days
S · 7
S · 7
F · 4
T · 4
W · 18
T · 4
M · 6
Last 4 weeks
50

如图所示，在 Karma [1] 展示页，**Todoist 通过展示任务所在的列表颜色来告诉你各个项目的任务完成比重**（收件箱中的任务默认为灰色）。所以根据这个特点，我们应该把顶级列表以及其下的列表全部都设为同一种颜色。比如我这里有三个主要颜色，**蓝色是个人**，**黑色是工作**，**红色是副业**。然后大家在图中一眼就可以看出周三这一天我完成了很多副业里的任务，这是因为周三这天我们播客发布新节目。

所以我建议在选择项目颜色的时候，要按照主类别，也就是顶级项目来区分颜色，每个顶级项目下的项目颜色要与顶级项目保持一致。

项目中任务的排列顺序

Todoist 的项目有两种排序方式：手动排序和按条件排序。其中按条件排序分为以下四种：

- 按日期排序
- 按优先级排序
- 按名称排序
- 按负责人排序

在 iOS 上和 macOS 上都可以手动排序，但是按条件排序只能在 macOS 端或者网页端才能执行。当执行过后，排序结果会自动同步到 iOS 端。

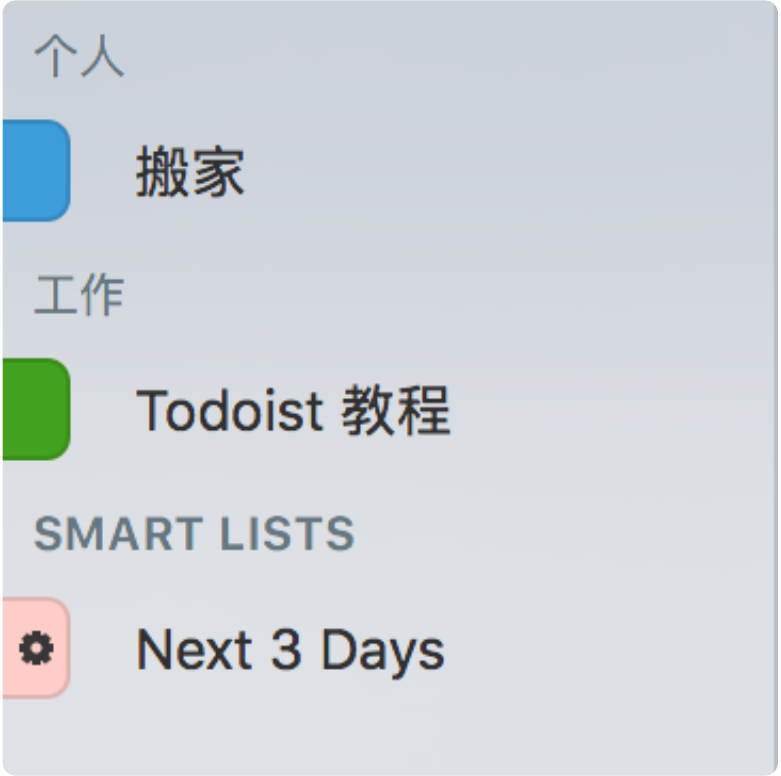
附：关于列表的无限嵌套这个需求

一些人特别向往任务管理工具在结构上提供「无限层级」，这个词说起来好像有一股魔力，看起来功能非常强大，但实际上就是列表的嵌套罢了。从操作上来看——尤其是用 OmniFocus——嵌套超过 3 个，在 iOS 上切换列表就已经非常麻烦了，更不要说「无限」了。

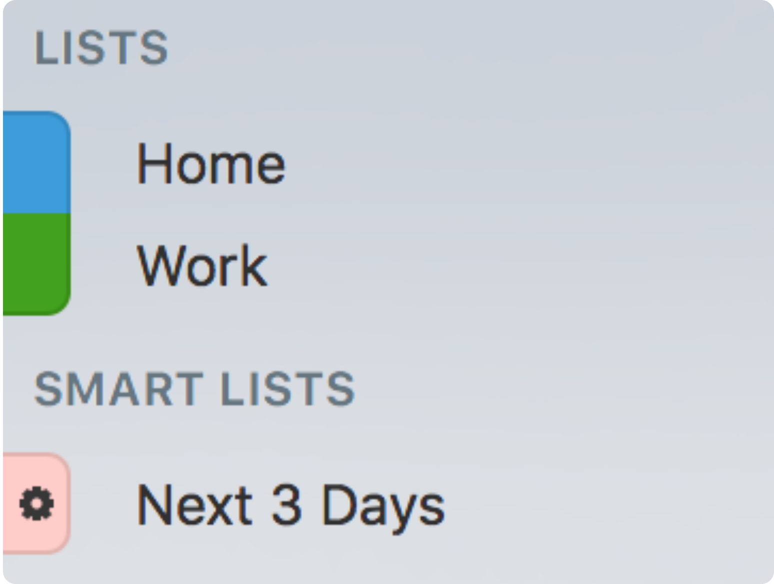
而 2Do 在这一点上的修正又有点过。在官方文档中明白地写着过多的任务层级会让任务结构增加不必要的复杂性[2]，所以 2Do 只给了三个层级：

- 列表（List）作为主项目
- 项目（Project）/清单（Checklist）作为子项目
- 项目或清单中的子任务作为最底层的任务

虽然在列表之上还可以建立列表组，但 2Do 的这个列表组从界面上就能看出来主要是起一个划线作用：


个人
搬家
工作
Todoist 教程
SMART LISTS
Next 3 Days

而且如果看 2Do 的默认层级划分，你可以发现，2Do 不想让我们用列表组的方式去划开任务，它希望正常情况下我们只用列表做顶级层级：


LISTS
Home
Work
SMART LISTS
Next 3 Days

Todoist 在 iOS 上的列表是折叠式的，所以切换列表时操作不会有 OmniFocus 那么复杂。同时，Todoist 对于列表层级或者说任务结构没有什么特殊定义，你可以尽管按你的需要去嵌套项目和任务。

但是我还是提个醒，列表这个形式，归根结底的目标是定位。那么针对这个目的，我们的要求就

是：**你可以不依靠搜索，仅通过列表层级就能快速定位每一个任务。**所以列表层级还是要尽可能得简洁。

标签

在任务管理中，**项目的意义在于层级定位，标签的意义在于任务所需的特殊情境。**在工具里，标签的具体功能是**横跨不同的项目，从中取出符合条件与当下执行情境的任务，再把它们集合在一起展示。**比如正文第一篇中提到的「<5mins [3] 」这个标签，也许我们很多列表里都有五分钟内可以完成的任务，通过这么一个标签，我们就可以把所有项目里五分钟之内可以完成的任务摘出来。

标签这个东西具体应该怎么用，根据不同的人可以有完全不同的答案，我只能尽我所能地提供一些思路。

日期和优先级

最重要的两个「标签」是**日期和优先级**。各个任务管理工具都从功能上都对它们予以重视，大多数做任务管理的人都会用这两个标签，而且任务管理的方法论都会提到它们。

在一个复杂的任务系统里，仅凭这两个标签就可以帮你直接抽出时间上和紧急程度上最优先的的事。只是在现代的任务管理工具里，因为可以赋予它们更合适的实现方式，所以它们都已经脱离

了标签的形式，化身为了具体的，可以被自动化的功能。

日期

范例：今天、三天内、一周内等。

关于任务管理有一句我很感同身受的话：

> If it doesn't get scheduled, it will never get done.

小专栏：为什么任务系统里的任务做不完不是问题?

想象一下你在工作中发现了流程上或者工具上有一些不顺手的情况，但当时因为你在工作所以没有办法腾出手立即解决它，只能先忍着完成工作。那么这时候，你可以跟同事抱怨一番第二天继续忍着老的流程和工具工作。也可以把这些问题记在任务管理工具中，回家调查解决方案或寻找更好的工具，去解决这个问题。

理论上，一个真正把任务管理系统用起来的人，他这个系统中的任务是做不完的。因为加一个任务只要不到一分钟的时间，而执行和完成这个任务要花的时间要多得多。而且如果一个人已经把使用任务管理工具养成习惯的话，他加任务的速度是会很快的。这不是一件坏事，因为任务管理系统就是为了让我们在想到一件事但无法做的时候，帮我们把它记下来。这件事以后做不做我们可以再判断，但是让工具帮我们记下它比我们硬记在脑子里要强，更比忘了它要强。

所以，任务系统里的任务增加速度快于解决速度是正常的，在这种情况下没有被加上日期的任务就非常容易被忽视。更不要提在团队的任务系统中，那些不断增多的任务里相当一部分甚至都不是我们自己加的，我们就更容易忽视它们。

在 Todoist 这样的现代工具里，任务的执行日期（包括时间提醒）已经化身为了具体的功能，我们已经不需要再手动地梳理任务，把今天要做的给单独整理出来。

Todoist 里的「今天」

在退出 Todoist 后重新打开它，你会发现默认的启动界面就是「今天」这个界面，Todoist 的通知中心部件显示的也是那些「今天」中的任务，Apple Watch 上的 App 也是如此。这充分地体现了

在任务系统中，首先关注今天要完成的任务是多么重要。Todoist 在有意培养用户的这个意识，我也是在用了 Todoist 之后真正意识到并做到「今日事，今日毕」的。

Todoist 的「今天」界面有以下几个特点：

特点一：在「今天」这个界面里，一个任务如果逾期未完成，就会出现在「今天」这个界面的最顶端。

特点二：提醒时间和日期在 Todoist 是区别开的，如果一个任务指定了提醒时间，但没有指定执行日期，那么它**不会出现**在「今天」这个界面中。

特点三：如果你与他人共享了一些列表，在「今天」里也不会显示其他人今天要完成的任务。

特点四： 如果你在「今天」这个界面直接添加任务的话，Todoist 会自动在添加任务时为这个任务附上今天的日期。

优先级（下一步行动）

在任务系统中，不管是安排了日期的任务还是没安排日期的任务，都在执行上有先后顺序和主次顺序。先后顺序虽然可以按提醒时间来排，但对于时间不那么具体的任务，就可以和主次顺序一样，通过优先级来排序。

因此任务管理系统中，优先级的重要性仅次于日期。而在现代任务管理工具里，优先级这个功能已经演化为了「旗标」或者「星标」这样的形式。

小专栏：优先级需要分几层？

关于优先级其实有个形而上的定义问题：优先级需要分几层？你会发现一些工具——比如 Todoist ——提供了 4 个层级的优先级。不少人因此产生定义上困扰，出现优先级层级使用上的混乱。这时候就要再提一下前面提到的那句话了：

> 实践意义高于形而上的定义。

优先级要分几层这个问题，再进一步地问一下，实际上要解决的是「当有过多的优先级层级时，我们如何判断每两个层级之间的界限」。比如 Todoist 有 4 个层级，我们怎么判断「第二优先级」和「第三优先级」之间的界限？答案是，具体问题具体分析——你只有手中有任务，你才能判断出它的优先级是不是比另一个更高。

所以首先，优先级层级问题必须在实践里观察。具体地说，是看任务在它所在的项目里的优先级的高低，然后从高到低地指定优先级。所以优先级一般不能跨出项目去对比，它只能同一项目内的其它任务对比。

我优先级的问题上更倾向于只用两个层级，这样就不会有任何定义模糊的问题。而且按定义来说，「优先级高」就是「先做」嘛：**要么是下一步要做的，要么是还没轮到做的**。所以也有工具就是只提供了一个优先级，比如 OmniFocus 就可以用旗标来表示优先度高。虽然 Todoist 给了 4 个层级，但也这不耽误我们按两个来用，你选一个喜欢的颜色，记住它的位列[4]，无视其他颜色就好。

并不是每个任务都要安排优先级，实际上大多数任务都不需要安排优先级，优先级这种东西，一定是越稀少越容易引起注意力，如果大多数任务都有优先级的话，那从视觉上就变成了一种单纯的颜色区分了，也就失去了它的指示意义。

地点、设备、人物

地点、设备、人物的重要性虽然要比日期和优先级差些，但也是传统标签思路的代表。「地点」和「设备」这两个标签的思路主要是为了完成批量操作，「人物」这个标签是为了便于追踪一件委托出去的任务，这些传统意义上的标签在现代工具上，也都有了多样的实现方式。

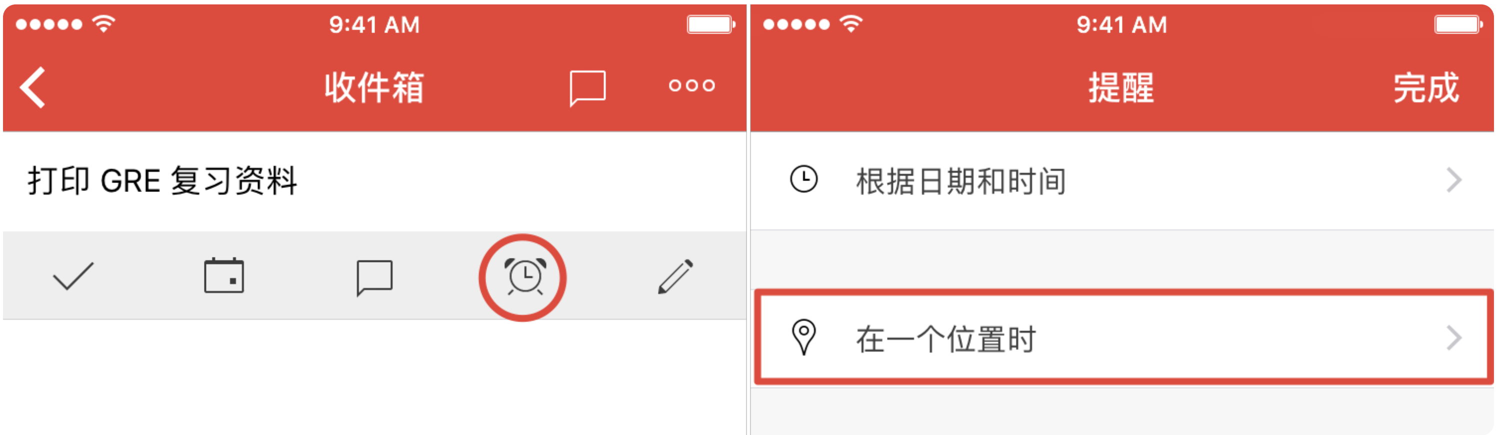
地点

在第一篇文章里我列出标签中，地点这个标签的范例有：办公室、家、实验室、健身房等位置。但是实际上我已经很少用地点做标签了。因为执行场景在家里的任务，一般会属于「个人」这个项目；执行场景在公司的任务，一般会在「工作」这个项目里；而执行场景在学校的任务，也会在「学校」这个项目里。也就是说，按照正常的分类思路，任务所处的项目和地点的关联已经很强，一般不会有存在于各个不同项目里的任务归属于同一个地点标签的情况。

在现代工具里，地点一般被考虑为一个提醒的条件，也就是到了预设的位置，就提示用户做某个预设的任务。Todoist 也实现了这一点，而且实现得和其它功能一样——自由度很高。

功能特点：

1. 可以设定「到达」或者「离开」某个位置
2. 对一个任务设定多个位置提醒
3. 保留地址记录，便于之后再次使用

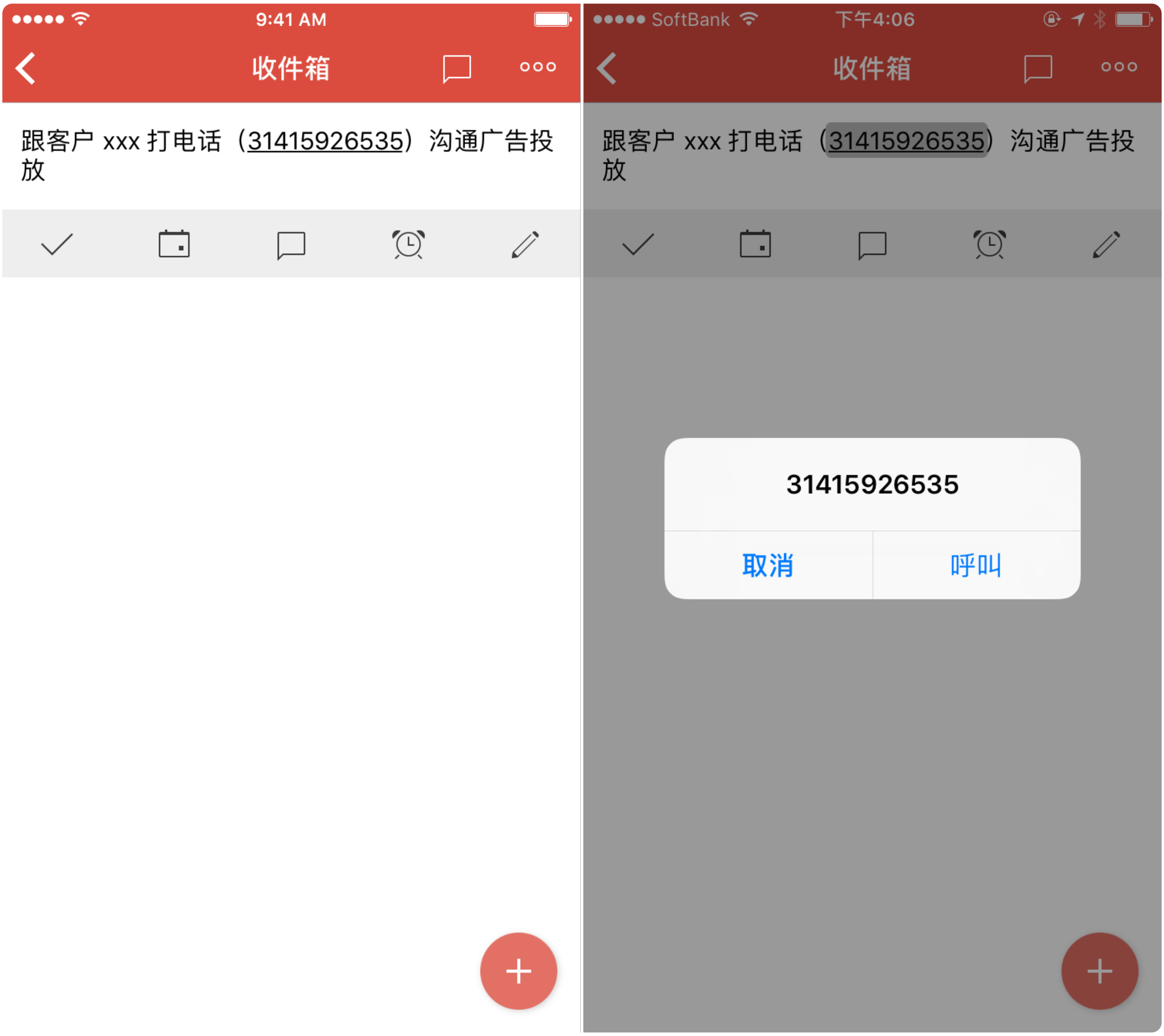

9:41 AM
收件箱
打印 GRE 复习资料
9:41 AM
提醒
完成
根据日期和时间
在一个位置时

位置提醒半径： Todoist 基于位置提醒的有效半径据我实测大概是 100m 的半径，这个半径不能进行调节。

设备

范例： PC、手机、画板等

「设备」这个标签，具体指的是完成任务所要用的设备，譬如要打的电话，可以标上 「手机」 这个标签，然后可以把要通过打电话执行的任务集中在这个标签下集中地处理。大多数的软件现在已经可以做到识别电话、URL 这样的功能，这可以让我们在执行相关任务时更迅速。



跟客户 xxx 打电话（31415926535）沟通广告投放

跟客户 xxx 打电话（31415926535）沟通广告投放

9:41 AM
收件箱
SoftBank
下午4:06
收件箱
31415926535
取消
呼叫

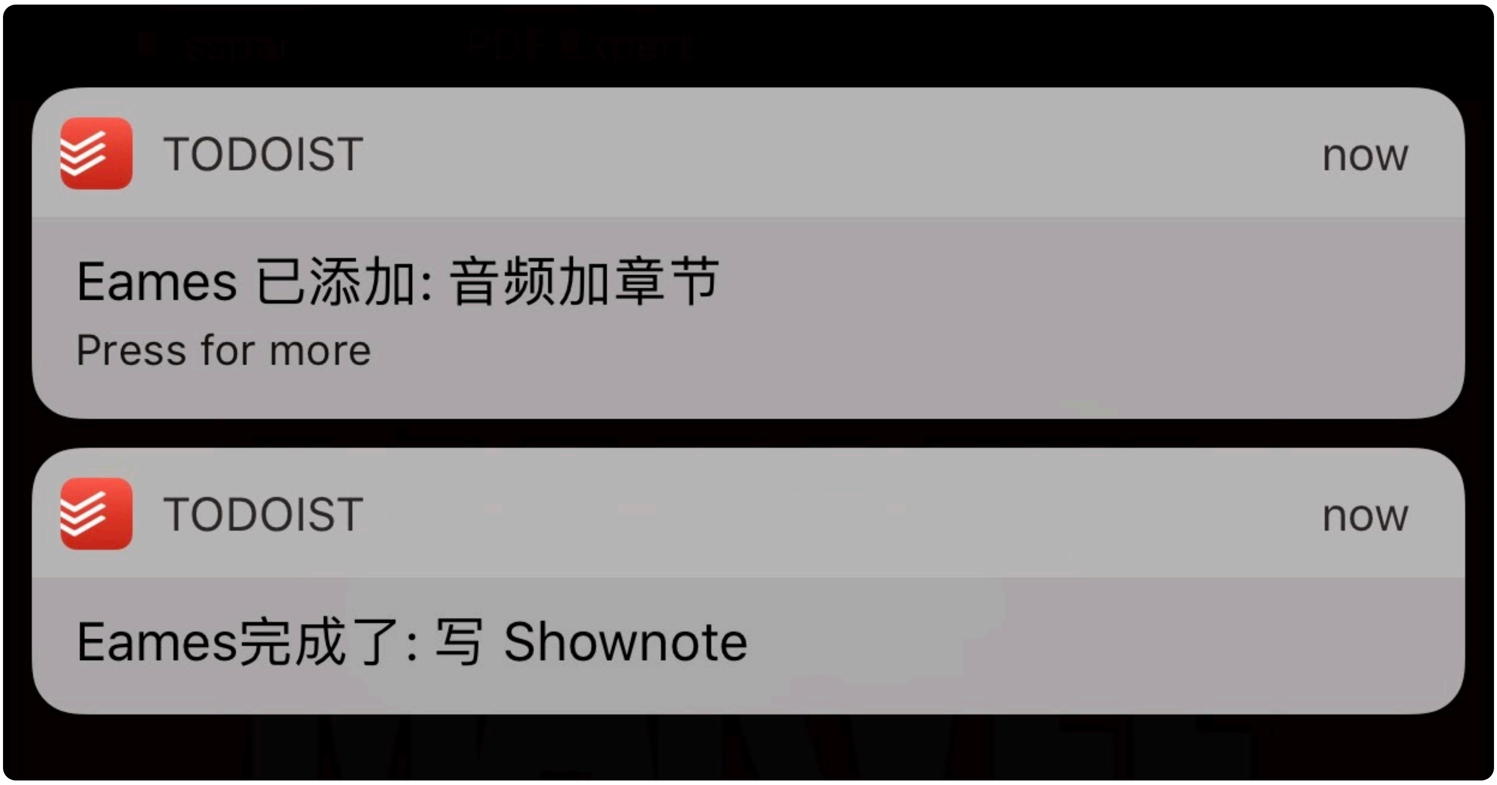
从图里我们可以看出 Todoist 会把长串数字识别为电话，在形式上变为可点的链接，点击后会出现是否呼叫的弹窗。

人物

有时候我们会发现一些任务并非独立的，而是一串任务中的一环，这些任务可能需要委托给别人，别人完成了他那部分工作以后再返回到我们这里，整串任务才能继续进行。这时候我们就需要对委托对象进行追踪，不然很容易忘记这个任务指派给了谁，进行到了什么底部，最终让这个任务迷散在任务系统中。

但对于真正意义上的大型团队协作级别的追踪委托对象和任务进度，个人用的任务管理工具并不能很好地胜任，所以这时候可以通过标签来标记出任务的委托人。

在 Todoist 里，如果你能说服对方也用 Todoist 的话，免费账户之间也可以共享任务，这时候你就可以体验 Todoist 中的任务指派功能。由于协作是后面才会写到的一个主要内容，所以在这个位置我只提一点它的好处：如果一个任务是你指派给对方的，那么默认情况下，对方完成这个任务时，你会收到提示：



Eames 已添加: 音频加章节

Press for more

Eames完成了: 写 Shownote

这个细节很大程度上减少了我和任务委托人的沟通成本。

一套有趣的标签

以上提到的标签，是传统方法论里都会提到的标签思路，所以这些标签，在现代工具里，或多或少地被改良成了更适合软件执行的方式，标签的痕迹逐渐不再明显。不过除了这些标签以外，还有一些标签思路是广大实践者们思考出的用法，它们在软件中无法功能化，所以以最传统的标签方式示人。

精力

初次接触按精力来划分任务这个想法是读了《A Fresh Take on Contexts》这篇文章，作者 Sven Fechner 在文章中按照任务的所需时间与注意力的结合，提出了一套有趣的标签：

- Short Dashes，指五到十分钟可以完成的任务，比如洗一下碗，买个新 App
- Brain Dead，指非常低迷的精神状态，一般会出现在每天最后的一段时间，但即使在这种状态下还是有不少事可以做。比如给文档分分组，标下标签；上传一些图片等
- Routines，指一些惯例上的事务，需要一定的精力和时间，有周期性。比如每周平一次信用卡的账，看看和自己记账软件里的结果是否一致等
- Full Focus，指需要长时间全神贯注的事情。写论文、做报告、做设计等等，一般一个职业最重要的工作大概都属于这个标签之下的任务。
- Hanging around，指时间长，又不需要太多精力的任务。比如读读文章，看一些纪录片之类的

按精力划分任务的思路非常有趣。毕竟我们是人，不能高强度地连轴转，精神状态有高有低，能找到合适的精神状态下做合适的任务是比仅仅通过时间来划分任务更合理的思路。

但在执行中，我却发现这一套东西我没有完全用起来，确切地说，我只使用了其中的一个标签——Brain Dead，去帮我把各项目里不需要什么时间和精力都能做的任务给提取出来。

没有用起来的原因首先是精神状态和任务本身的相关性很强，像写这篇文章，就算不标记 Full Focus，我也是在 Full Focus 的状态下写的；其次是从操作上来说，给每个任务去标记一个精神状态的标签成本实在是太高了。

所以我对这套标签的思路接近我对优先级的态度——在优先级那部分，我只取了其中一个优先级，在其下的任务，优先级就是最高的，要下一步做的。在精神状态这里，我只取一个精神状态「Brain Dead」（我的这个标签叫「EasyDo」），在这个标签下的，就是那些不用动脑子特别快就能搞定的任务。

本节小结

这一节的重点是：我们要把很多在现代工具中已经转化为特殊功能的那些标签重新以标签这个概念来把握。因为如果我们把它们看为具体的功能，就没办法将任务管理工具的各种概念给系统化，就会在选择和把握工具时，又陷入无重点的功能对比。

此外，除了我提到的这些标签范例，其实还有非常多的各式各样的标签，出于实用性和篇幅的考虑，我没有把我知道的所有标签思路都写进来。实际上，我个人在 Todoist 中对标签的使用非常少，原因我想是因为主要标签都被转化为了具体功能。

过滤

过滤
是现代任务管理工具的先进之处

过滤，是现代任务管理工具的先进之处，意义所在。

在知乎 Live 里和前面的文章中，我会时不时地把标签和项目称为「维度」，这个维度是什么的维度？是过滤的维度。

什么是过滤?

1. 过滤是高级任务管理工具和基础任务惯例工具的分界线。
2. 我对过滤的定义是：**设定一组或多组条件，匹配出系统里符合条件的个体，将它们聚集在一起。**
3. 任务管理系统中的过滤则是：通过项目和标签（以及标签的各种形式），交叉定位系统里符合要求的任务，将这些任务汇集到一处。

为什么需要过滤?

任务管理不光是一套储存任务的系统，它更重要的意义是在合适的情境把合适的任务提供给我

们，换句话说是让我们在看的时候只看到我们当时需要的信息。在这个层面上，标签和项目就是任务管理的储存部分，而任务管理真正的查看部分应该是过滤，过滤才是帮你按照理想要求**汇集**任务的地方。

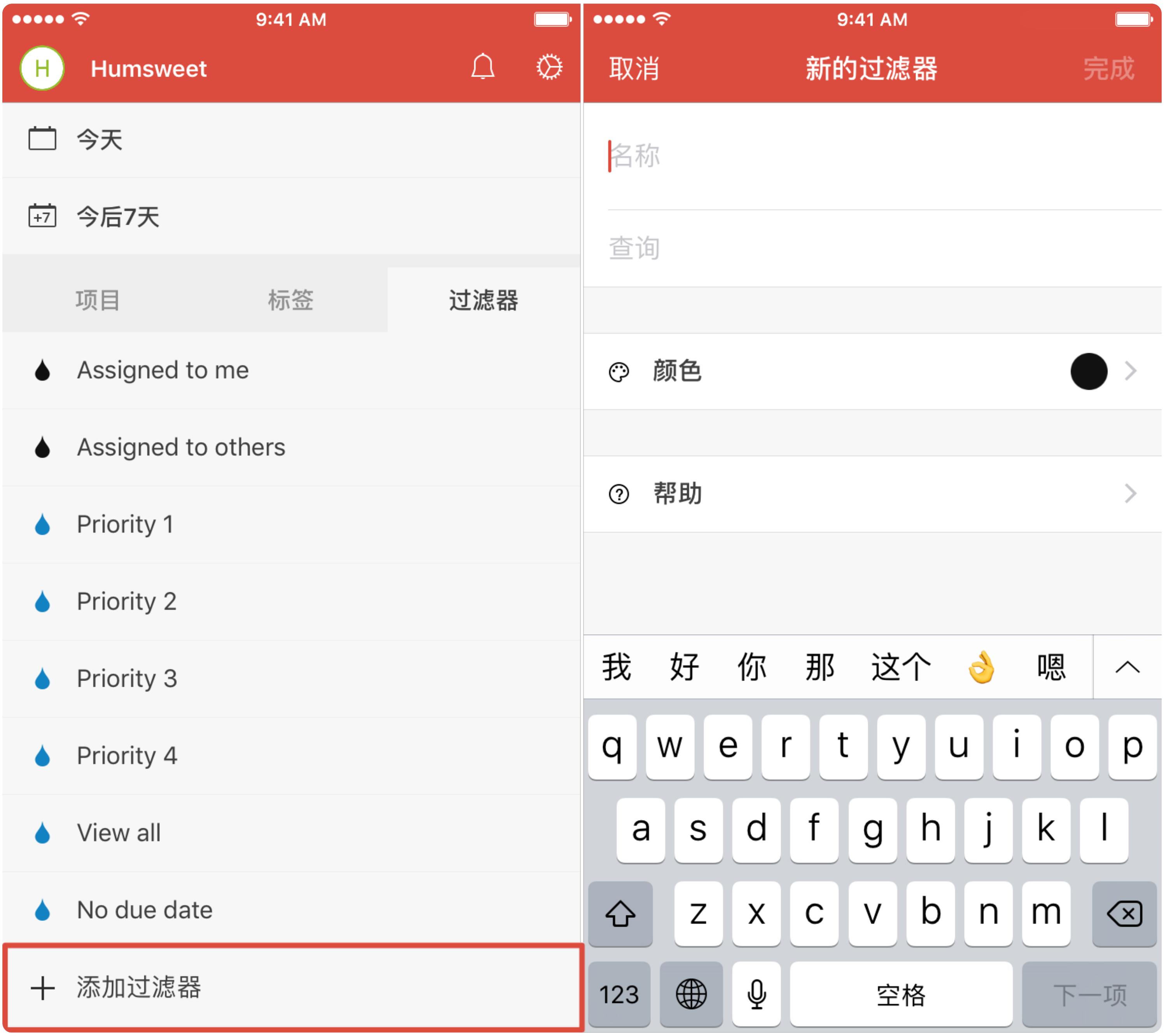
可以说，如果一个任务管理系统稍显复杂，通过标签和项目就还是不能很好地把我们需要的信息展示出来。譬如说一个小团队的成员，每个人在查看任务的时候，理所应当的是看到自己今天在工作中需要完成的任务，而通过项目，他就不可避免地看到团队其它人的任务；而通过「今天」这个标签，他又会不可避免地看到所有今天要做的任务，不管它们是不是在「工作」这个项目里。

但是把「今天」这个标签和「工作」这个项目放在一起，再加上一个指派给我的属性，就能解决这个需求。

最后，虽然在这里会给大家一种项目和标签相较于过滤有点弱的感觉，但好的储存系统是制作过滤的地基。制作一个好的过滤的前提，也需要自己对自己这个系统足够了解，这个了解的标准是不需要通过查找就能通过自己的项目层级快速定位任务。如果一个任务管理系统的列表和标签搞得很乱的话，就无法做到这一点，进而也就很难做出一个有效的过滤。

Todoist 里添加过滤

在 Todoist 的主界面上有三个分页，其中最后一个就是过滤的功能，Todoist 的中文翻译是「过滤器」。添加一个过滤器的方法是在列表底部找到「+添加过滤器」，然后输入过滤器的名称、语法。


9:41 AM
Humsweet
今天
今后7天
项目
标签
过滤器
Assigned to me
Assigned to others
Priority 1
Priority 2
Priority 3
Priority 4
View all
No due date
添加过滤器
9:41 AM
取消
新的过滤器
完成
名称
查询
颜色
帮助
我 好 你 那 这个 👌 嗯
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
空格
下一项

Todoist 中的过滤通过特殊的语法来实现，这套语法虽然支持中文，但还是推荐用英文。这套语法的文档有 中文版本 也有 英文版本，但英文版本明显要比中文版本的全很多。

一些人发现在 iOS 端的 Todoist 没有搜索功能，这一点可以借助过滤器来实现，语法是 `search: 关键字` 。通过这种方法你可以搜到所有在标题或者评论中含有这个关键字的任务。

单维度过滤

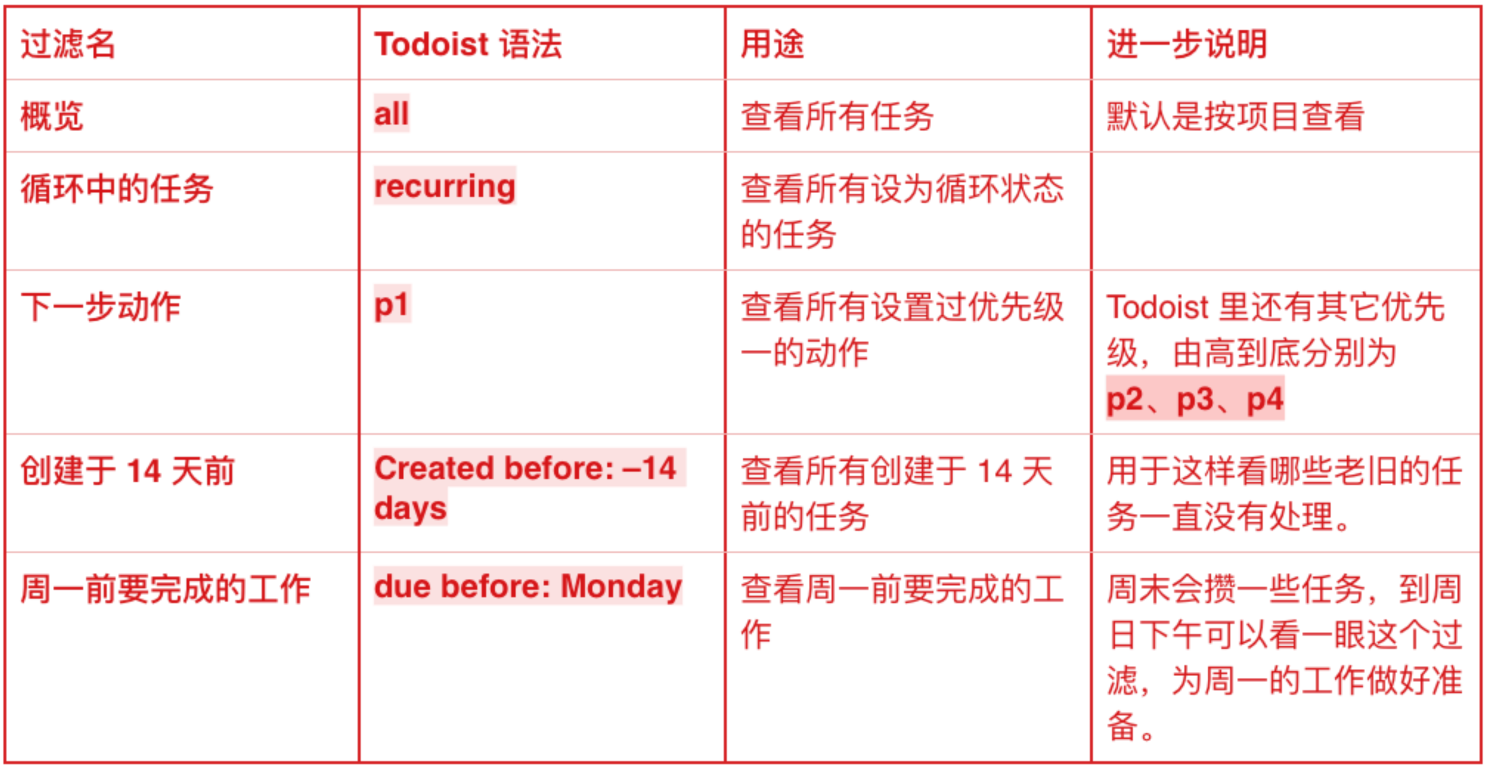
定义：仅通过一个维度形成过滤。

基本语法：

- `@` ： 指标签，比如 `@BrainDead` 就是过滤出包含 @BrainDead 这个标签的任务
- `#` ： 指项目，但是只显示该项目下的所有任务，**不包括该项目下的子项目**。比如你的 Work 项目下有三个子项目，过滤中输入 `#Work` 的话就只能显示 Work 项目本身的任务，不包括它下面三个子项目中的任务
- `##` ： 指项目，显示该项目与子项目下的所有任务。还是比如 Work 项目下有三个子项目，这时候在过滤中输入 `##Work` ，就不仅能显示 Work 项目下的任务，也能显示下面三个子项目中的所有任务。这个语法也可以用 `p:项目名` 来代替。

单维度过滤在用法上，有一定的正误之分。所谓错误用法，是单独把某一个项目或者某一个标签作为过滤。这种行为有点类似把一张专辑放在了一个播放列表里，而且这张播放列表只有这一张专辑……

而正确的用法是指通过现代工具的特性，将那些符合某一个标准的任务聚集在一起。以下是一些 Todoist 中用得上的单纬度标签示例：



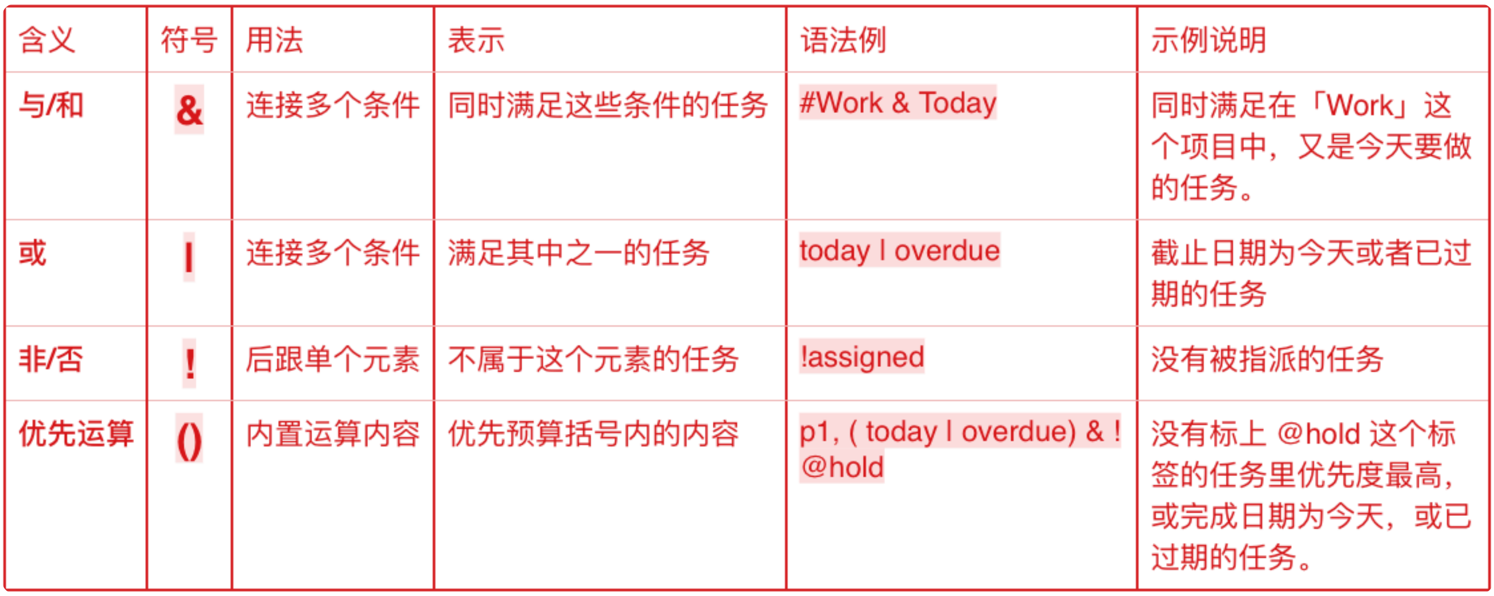
过滤名	Todoist 语法	用途	进一步说明
概览	all	查看所有任务	默认是按项目查看
循环中的任务	recurring	查看所有设为循环状态的任务	
下一步动作	p1	查看所有设置过优先级一的动作	Todoist 里还有其它优先级，由高到底分别为 p2、p3、p4
创建于 14 天前	Created before: -14 days	查看所有创建于 14 天前的任务	用于这样看哪些老旧的任务一直没有处理。
周一前要完成的工作	due before: Monday	查看周一前要完成的工作	周末会攒一些任务，到周日下午可以看一眼这个过滤，为周一的工作做好准备。

单纬度过滤示例

复杂过滤（多维度过滤）

定义：运用项目、标签、日期、委托人等多种任务维度来制作过滤。

在前面我举了一些我自己在用或用过的单维度过滤的范例。其实如果大家看 语法的文档 的话，会发现在 Todoist 里这样的语法有非常多，只不过大多数这样的单个元素实用性不高，需要在更复杂的组合下才能发挥出更好的作用。而在文本输入模式中组合不同的元素形成筛选条件，相信你已经意识到我们下一步要谈的就是逻辑运算符。

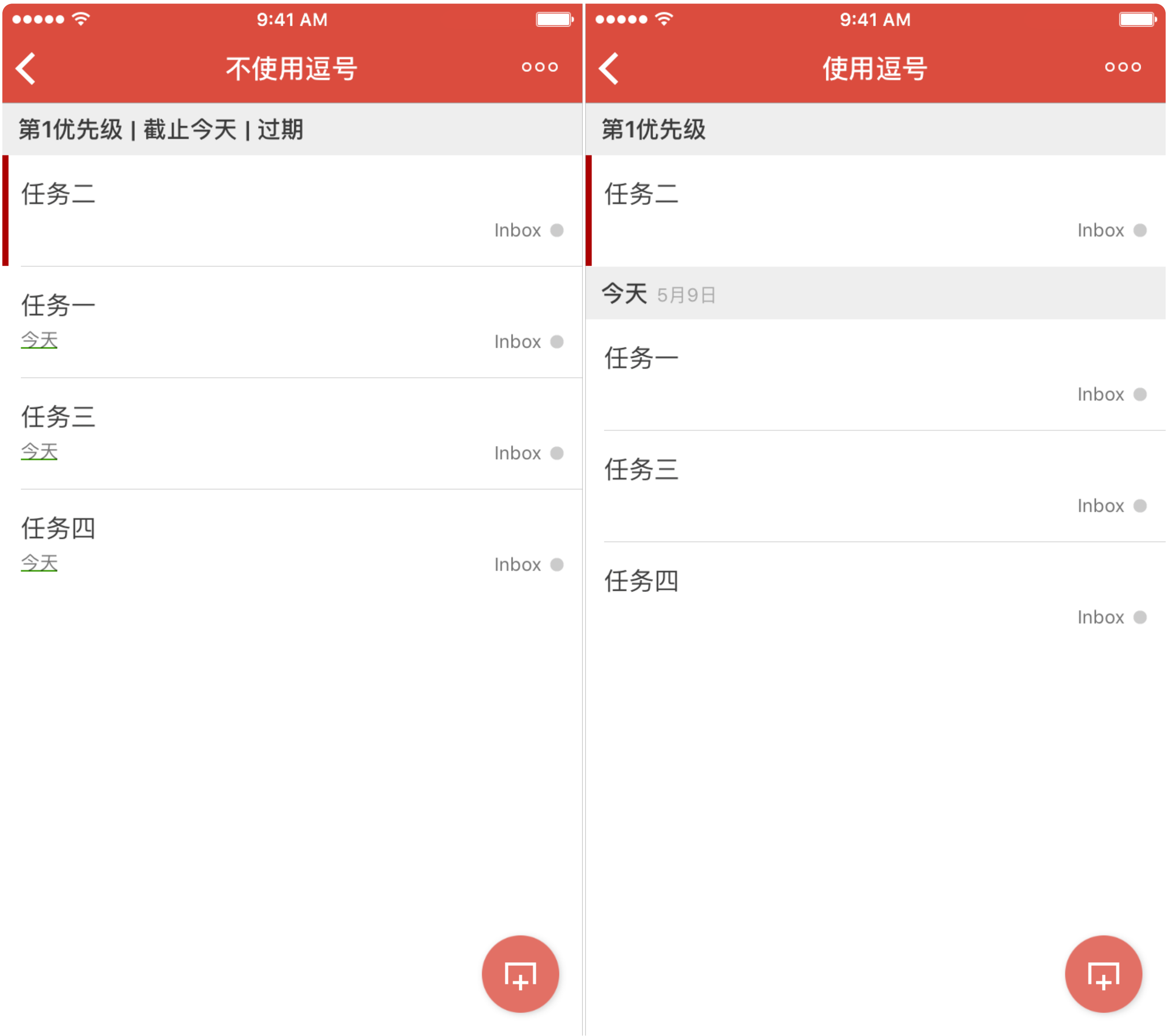


含义	符号	用法	表示	语法例	示例说明
与/和	**&**	连接多个条件	同时满足这些条件的任务	`#Work & Today`	同时满足在「Work」这个项目中，又是今天要做的任务。
或	**|**	连接多个条件	满足其中之一的任务	`today | overdue`	截止日期为今天或者已过期的任务
非/否	**!**	后跟单个元素	不属于这个元素的任务	`!assigned`	没有被指派的任务
优先运算	**()**	内置运算内容	优先预算括号内的内容	`p1, ( today | overdue) & ! @hold`	没有标上 @hold 这个标签的任务里优先度最高，或完成日期为今天，或已过期的任务。

多维度过滤中的运算符

用半角逗号并联过滤

除了组合多个元素形成过滤，Todoist 还提供了方法让我们组合多个过滤形成一个大过滤。它的效果主要是做到过滤结果的分隔显示，比如后面提到的「真•下一步动作」，它的语法是 `p1, ( today | overdue) & !@hold`，我们其实可以把 「p1」 写进括号里面—— `( p1 | today | overdue) & !@hold`，但这样所有任务会挤在一起，视觉上很难受。考虑到按优先级的部分和按时间的部分还是有一定的区别，所以我们理论上可以把它们从视觉上分开，实践起来效果也不错。



第1优先级 | 截止今天 | 过期

任务二 Inbox

任务一 今天 Inbox

任务三 今天 Inbox

任务四 今天 Inbox

第1优先级

任务二 Inbox

今天 5月9日

任务一 Inbox

任务三 Inbox

任务四 Inbox

可能你已经看出来了，并联过滤的特殊符号是半角逗号 `,` 。只要用半角逗号隔开多组过滤规则，在显示上就会有一个明显的区别。

多维度过滤示例

一、真 · 下一步动作

语法： `p1, ( today | overdue) & !@hold`

解释：过滤出没有标上 `@hold` 这个标签的任务里，或优先度最高，或完成日期为今天，或已过期

的任务。并且以分栏显示。

下一步动作从传统的思路上是通过某种方式，给任务标一个特殊的优先级。但是实际上对于一个人来说，过去没完成的、今天要做的、以及优先级高的任务都在某种程度上是他的「下一步动作」，所以我们就可以把这些任务组合到一期。而且这些动作还不能包括那些因为特殊原因被暂停的任务，所以要排除掉 `@hold` 这个标签。同时为了显示上更清晰，我把这个过滤的优先度部分和时间部分通过 ， 进行分开展示。

二、真 · 创建于 14 天前的动作

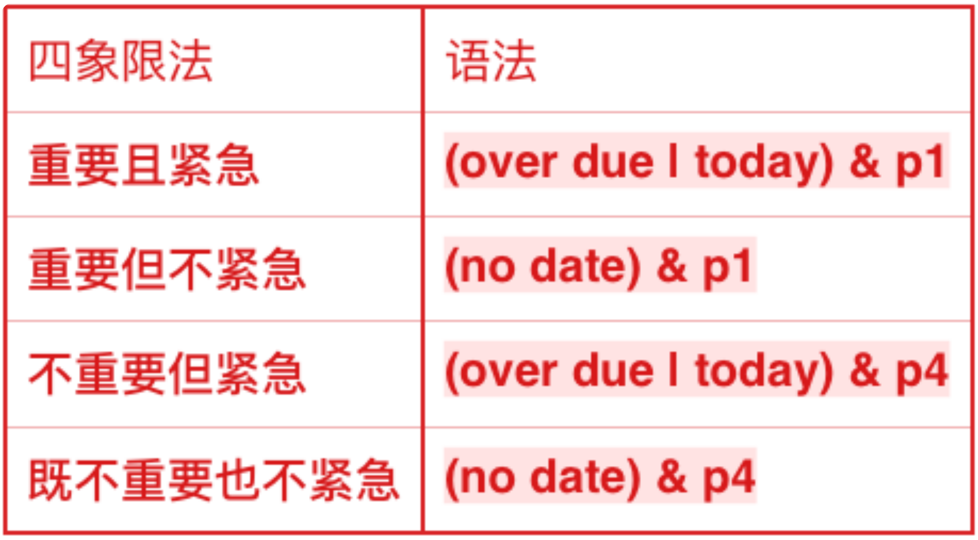
语法： `created before: -14 days & !recurring`

解释：创建于 14 天前，并且不在循环状态下的动作。

相信看了解释大家就会明白，如果你只用 `created before: -14 days` 这一个语法的话，它会在过滤结果中出现那些循环中的任务。而循环中的任务，我们只要用得上，并不用在乎它是多早之前建立的。所以为了避免看到这种我们不需要注意的任务，我们从语法上要把它排除掉。

三、四象限法

四象限法是一种根据事情的时间和优先度来划分任务的方法，这套方法非常适合通过过滤来实现。



四象限法	语法
重要且紧急	(over due | today) & p1
重要但不紧急	(no date) & p1
不重要但紧急	(over due | today) & p4
既不重要也不紧急	(no date) & p4

单纬度过滤示例

Todoist 中过滤的排序框架

Todoist 目前不提供用户自定义的排序功能，他们表示考虑以后会把这个功能加进来[5]。

根据我的观察，Todoist 目前的排序框架有三个特征：

- 如果过滤里加入了明显的时间语法，比如 `-3 days`，它就会按照日期来排列
- 如果过滤里没有这样的语法，则会先把设有优先度的任务放在最上方，其后跟着把带有执行时间的任务。不管是有或是没有优先度/执行时间的任务，它们会先按照项目组织在一起，在项目内再按标签组织在一起
- `all` 这个语法是个例外，它是按照项目来排列所有内容的

---

1. Karma 系统是 Todoist 的一个简单的奖惩系统，任务完成得多就加分，少就扣分。分数多到一定地步会升级，升级到特定等级后——比如「专业人士」「专家」——Todoist 都会给用户送出两个 3 个月的高级账户兑换码，一个可用于自用，一个可用于赠送。↩
2. 原文：2Do, for valid reasons, does not provide support for sub-projects as such a feature would in fact add needless complexity to the overall structure of your tasks…… ↩
3. 原文是「不到 5 分钟行动」，但这个标签名实在太憨了…… ↩
4. 红色是 p1，灰色是 p4，这在做过滤的时候有用。 ↩
5. 在这篇文章的评论下有提到：https://blog.todoist.com/2017/02/23/todoist-filters-update/ ↩ ↩

102

上一期
第一章：Todoist 如何满足 GTD

下一期
第三章：使用 Todoist 进行任务...

评论将在经过作者筛选后公开展示

发送

精选评论（0） 我的评论

支持我们 作者招募 用户协议 FAQ Contact Us

© 2013-2018 少数派 | 粤ICP备09128966号-4 | CC BY-NC 4.0